REMARKS

Claims 1-20 are pending. Claims 1-18 are rejected. Claim 16 has been amended. Claims 19 and 20 have been added. No new matter has been added as a result of this amendment.

102 Rejections

Claims 1-18 are rejected under 35 U.S.C. § 102(b) as being anticipated by Dey et al. (U.S. Patent No. 5,513,123). The Applicants have reviewed the cited reference and respectfully submit that the present invention as is set forth in Claims 1-18 is not anticipated or rendered obvious by Dey et al. (U.S. Patent No. 5,513,123).

The Examiner is respectfully directed to independent Claim 1, which recites that embodiments of the present invention are directed to:

> ... determining sets of mutually exclusive operations from the specified processor operations based on the desired instruction level parallelism; programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit ...

Independent Claim 10 recites limitations similar to those of independent Claim 1. Claims 2-9 and Claims 11-15 depend from Claims 1 and 10 respectively and recite further limitations of the claimed invention.

Dey et al. does not anticipate or render obvious a method for the automatic design of a processor datapath including "determining mutually exclusive operations from the specified processor operations based on the desired instruction level parallelism" as is recited in Applicants' Claim 1. Dey et al. only shows a non-scan design for testability of RT level data paths. It should be appreciated that the Applicants invention as set forth in Claims 1 and 10 include limitations that define a relationship between the recited mutually exclusive operations that are determined from the recited specified processor operations, and the recited desired instruction level parallelism. Namely, that the mutually exclusive operations that are

determined from specified processor operations be based on the desired instruction level parallelism. More specifically, that from an input specification (see Applicants' Claim 1) that may include register file specifications, processor operations and desired instruction level parallelism a set of mutually exclusive operations are derived. By contrast, the Dey et al. reference teaches that the input to his algorithm for designing datapaths is a "target datapath" (see column 15, lines 60-67). It should be appreciated that contrary to the assertions made in the outstanding Office Action, Figure 2b shows that according to Dey et al. operations may be transmitted in parallel but does not suggest that desired "instruction level" parallelism as a specified input be a factor in determining a set of mutually exclusive operations. In fact, nowhere in Dey et al. is it taught or suggested that from input specifications that specifically include desired instruction level parallelism is there derived a set of mutually exclusive operations. Consequently, Dey et al. does not anticipate or render obvious the Applicants' invention as is set forth in Applicants' Claims 1 and 10.

The Examiner is respectfully directed to independent Claim 16, which recites that embodiments of the present invention are directed to a method for synthesis of a register file including "determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the operations." Claims 17 and 18 depend from Claims 16 and recite further limitations of the claimed invention.

Dey et al. does not anticipate or render obvious a method for the automatic design of a processor datapath including determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the operations as is recited in Claim 16. Dey et al. only shows a non-scan design for testability of RT level data paths. Dey et al. teaches at column 9, lines 55-65, that a register file may send and receive data from an EXU (e.g., execution unit) as a part of its data transferring operations but does not show or suggest that desired "instruction level" parallelism as a specified input be a factor in determining how a register port is shared as is recited in Claim 16. Nowhere in the Dey et al. reference is there disclosed this recited correlation between the sharing of a

register port for functional units and the specification of instruction level parallelism. Consequently, the Applicants' invention as is set forth in independent Claim 16 is neither shown nor suggested by Dey et al.

The Applicants respectfully submit that the rejection of Claims 1-18 under 35 U.S.C. § 102(b) was in error and respectfully requests the withdrawal of this rejection. The Examiner is reminded that in order to anticipate a Claim, the reference must teach each and every element of the Claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed Cir. 1987). It is clear from the discussion above that "each and every element" is in fact not described by the Dey et al. reference. Dey et al. does not "either expressly or inherently" describe "determining mutually exclusive operations" or "determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the operations" as was discussed above.

Therefore, Applicants respectfully submit that Dey et al. does not anticipate or render obvious the present claimed invention as recited in Claims 1, 10 and 16, and as such, Claims 1, 10 and 16 are in condition for allowance. Accordingly, Applicants also respectfully submit that Dey et al. does not anticipate or render obvious the present claimed invention as is recited in Claims 2-9 dependent on Claim 1, Claims 11-15 dependent on Claim 10, and Claims 17-18 dependent on Claim 16, and that Claims 2-9, 11-15 and 17-18 traverse the Examiners basis for rejection under 35 U.S.C. 102 as being dependent on an allowable base claim.

Conclusion

In light of the above-listed remarks, Applicants respectfully request allowance of the remaining Claims.

The Examiner is urged to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Respectfully submitted,

WAGNER, MURABITO & HAO LLP

Dated: 7/11, 2003

John P. Wagner, Jr.
Registration No. 35,398
Two North Market Street
Third Floor
San Jose, CA 95113
(408) 938-9060

CLAIMS

VERSION WITH MARKINGS TO SHOW CHANGES MADE

16. (Amended) A method for automatic synthesis of a register file and functional-unit register file interconnect in a processor, based on an input specification of register file types in the processor, specified processor operations, desired instruction level parallelism among specified operations and functional units in the processor,

the method comprising:

for each type of register file specified in the processor, establishing a set of read/write port requests between the functional units and each of the register file types;

programmatically computing a resource allocation of register ports in the register file types to read/write port requests, including determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the specified processor operations; and

programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports.

Please add the following new Claims:

19. (New) The method of Claim 16 wherein said input specification comprises a desired set of machine operations together with an abstract specification of concurrency and resource sharing constraints.

20. (New) The method of Claim 19 wherein a concurrency constraint identifies which operations are allowed to be issued at the same time, while a resource sharing constraint identifies which operations cannot be issued at the same time.